

Prozedurale Generierung von Baumstrukturen innerhalb der Unreal Engine 4

Bachelor Abschlussarbeit

David Liebemann, TODO

Überblick

1. Einleitung
2. Lindenmayer-Systeme
3. Space Colonization Algorithmus
4. Implementierung
5. Ergebnisse
6. Zusammenfassung und Ausblick
7. Quellen

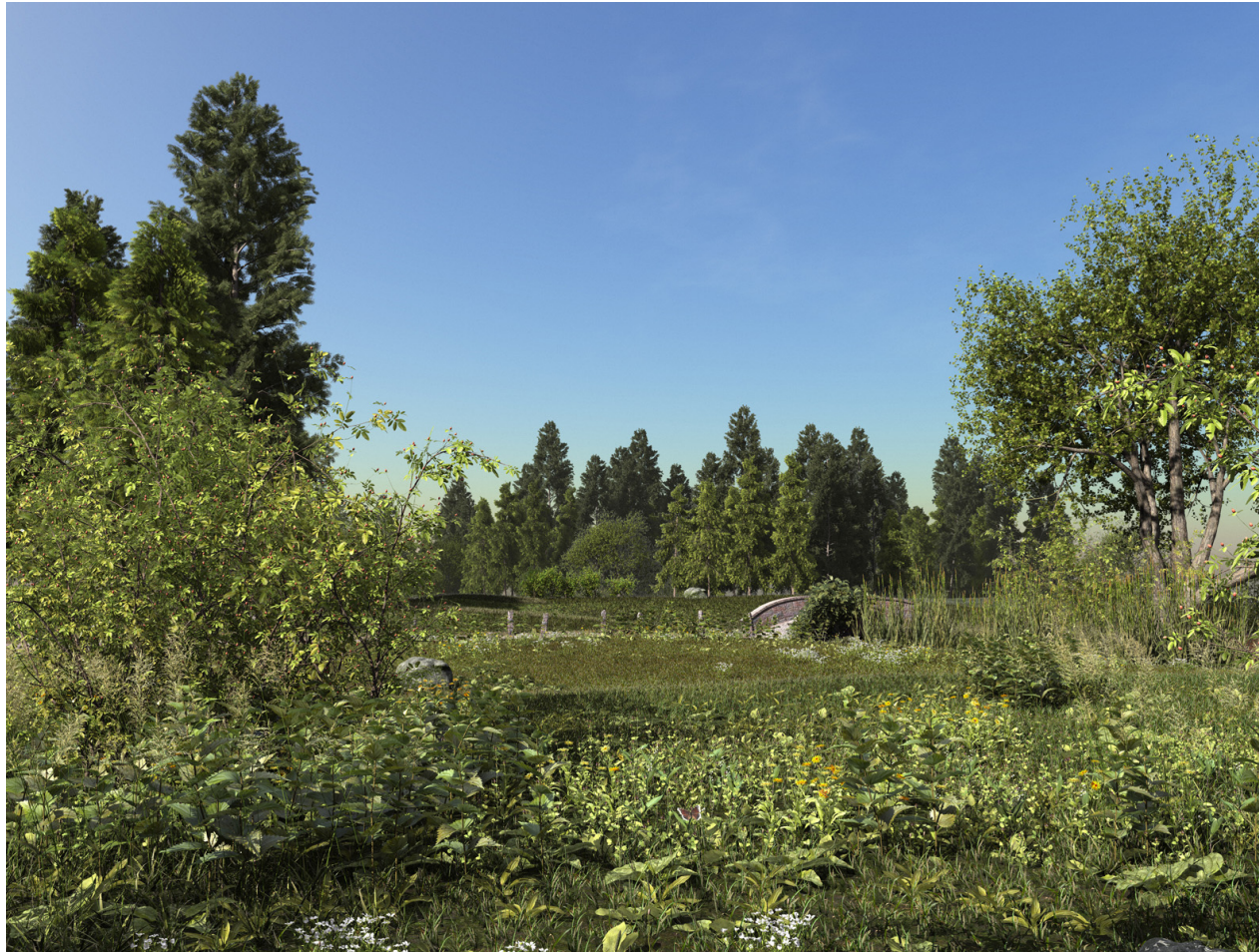
1. Einleitung

1 Einleitung

Prozedurale Generierung

- Konstruktion von 3D-Modellen durch computergenerierte Daten
- Benötigt eingeschränkten Eingriff durch Benutzer
- Generierung von Pflanzenmodellen ist ein wichtiger Bestandteil
- In dieser Arbeit: Konzentration auf die Generierung von Baumstrukturen

1. Einleitung - Prozedurale Generierung



Prozedural generierte Landschaftsszene. [Gre]

1. Einleitung - Ansatz

Ansatz

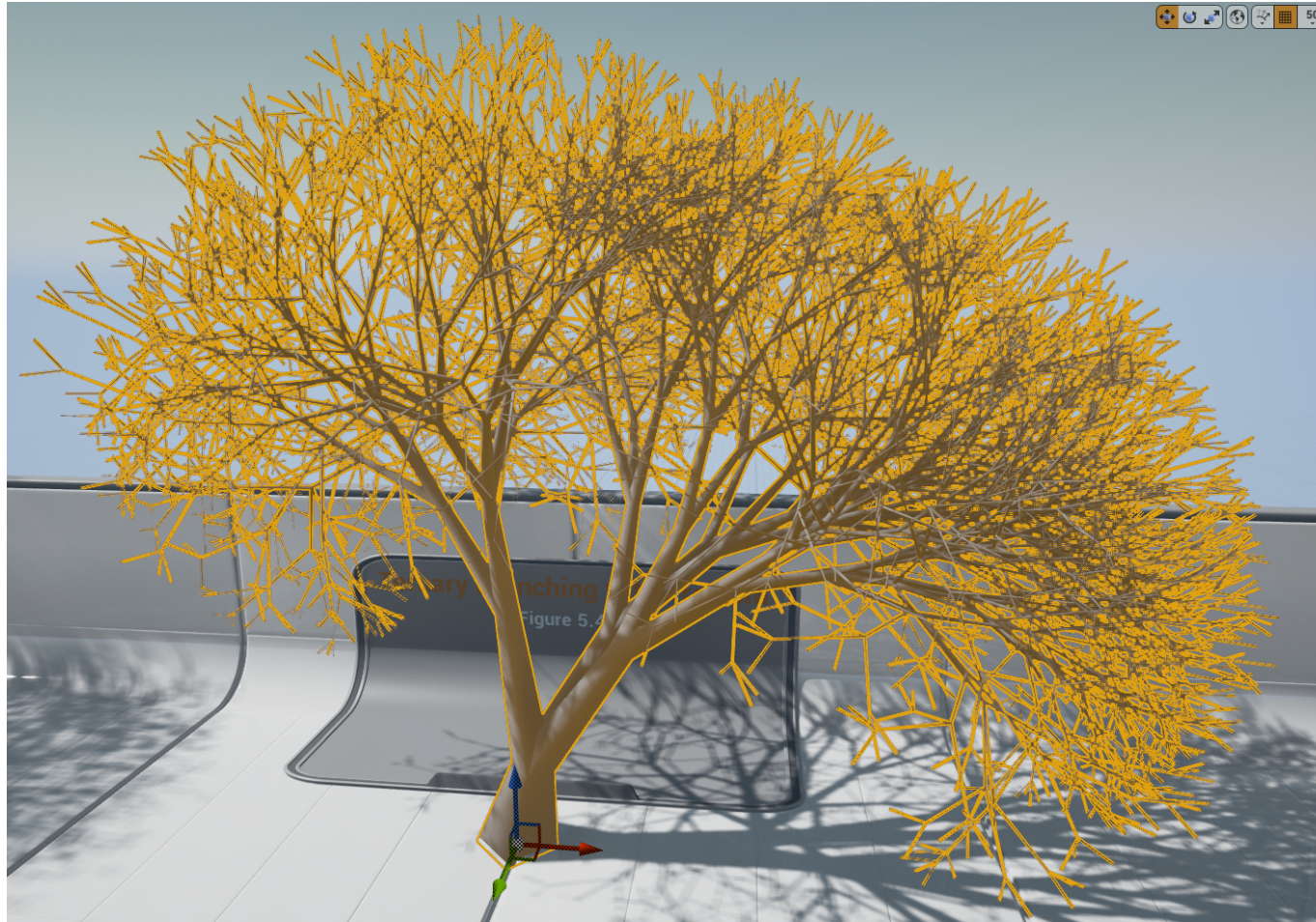
- Implementierung von zwei Verfahren zur prozeduralen Generierung von Baumstrukturen:
 - Lindenmayer-Systeme
 - Space Colonization Algorithmus
- Verwendung des Frameworks der Unreal Engine 4
- Vereinfachte Darstellung von Ästen in Form von Zylindern

1. Einleitung - Unreal Engine 4

Unreal Engine 4

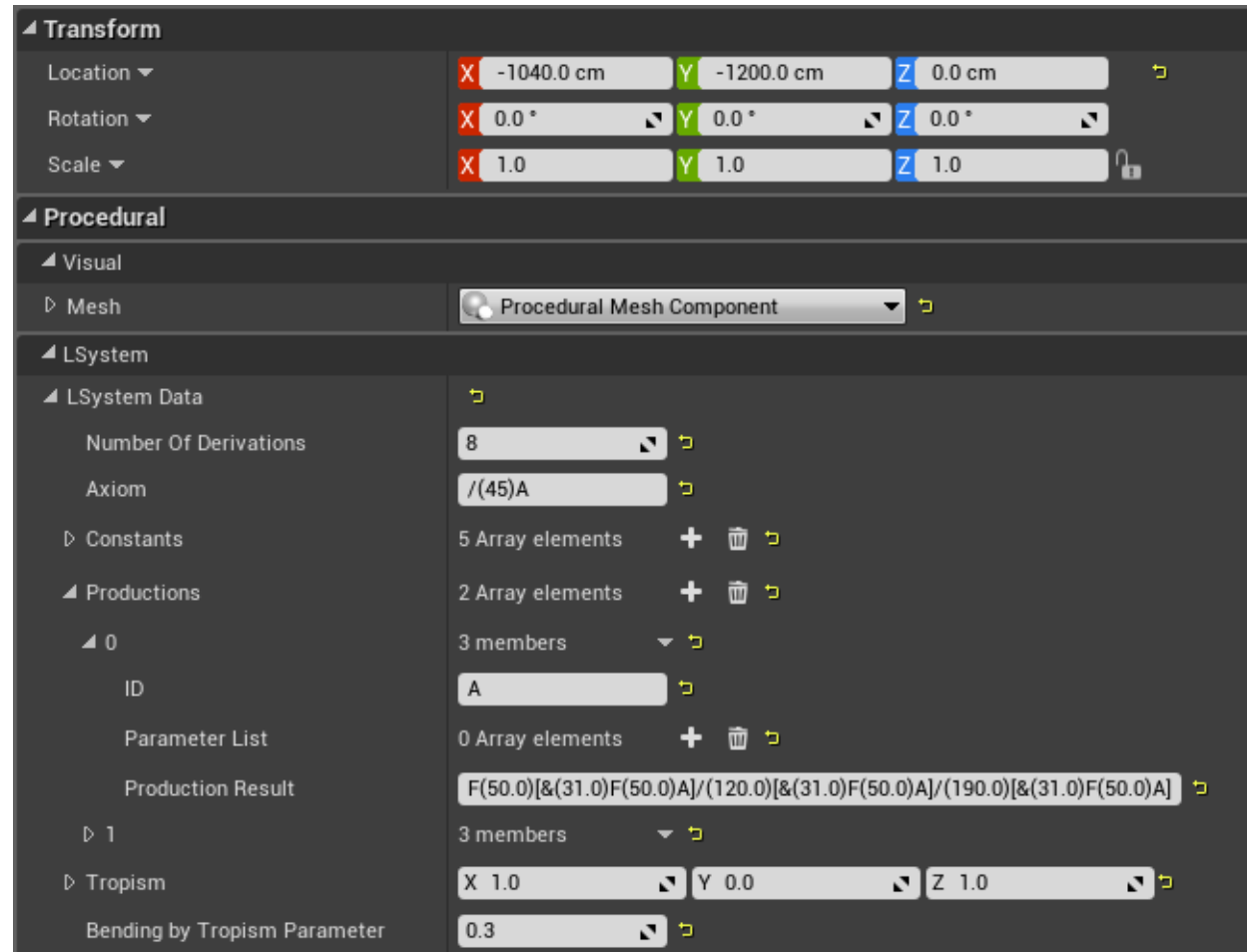
- Sammlung von Softwarewerkzeugen
- In C++ programmiert mit frei zugänglichem Quellcode
- Inhalte werden in C++ oder Blueprint erstellt und leiten von Framework-Basisklassen ab
- Verfügbarkeit eines visuellen Editor:
 - Ermöglicht die einfache Positionierung von Actors
 - Erlaubt die Eingabe von Parametern über das Editor-UI

1. Einleitung - Unreal Engine 4



Positionierung eines Actors im Leveleditor

1. Einleitung - Unreal Engine 4



Eingabefenster für Actor-Parameter

2 Lindenmayer-Systeme

- Von Aristid Lindenmayer 1968 entwickelte Erweiterung von Ersetzungssystemen
- Weitere Ergänzungen durch Prusinkiewicz und Lindenmayer in 1990
- Funktionsweise basiert auf der Ersetzung von Zeichen in Zeichenketten
- Grafische Interpretation der Resultate ergibt Modelldaten

2. L-Systeme - D0L-Systeme

2.1 D0L-Systeme

Definition D0L-System: 2.1 *Ein deterministisches, kontextfreies L-System (D0L-System) ist ein Tupel $G = (V, P, \omega)$, bestehend aus:*

V *Einem nichtleeren, endlichen Alphabet.*

P *Einer endlichen Menge von Produktionsregeln in der Form $P : a \rightarrow b$ mit $a \in V$ und $b \in V^*$.*

$\omega \in V^+$ *Dem Axiom, Startwort des L-Systems.*

2. L-Systeme - D0L-Systeme

Verwendete Begriffe:

Deterministisch: Es existiert genau eine Produktionsregel für alle Symbole in V

Kontextfrei: Ersetzung findet unabhängig von umgebenden Symbolen statt

Ableitung: Gleichzeitige Ersetzung aller Symbole eines Wortes anhand der Produktionsregeln

2. L-Systeme - D0L-Systeme

Beispiel: Simulation des Wachstums der Blaualgen-Gattung „Anabaena“

V besteht aus den Symbolen $\{a_l, a_r, b_l, b_r\}$

a und b : Größe und Teilungsbereitschaft einer Zelle

l und r : Zellenpolarität

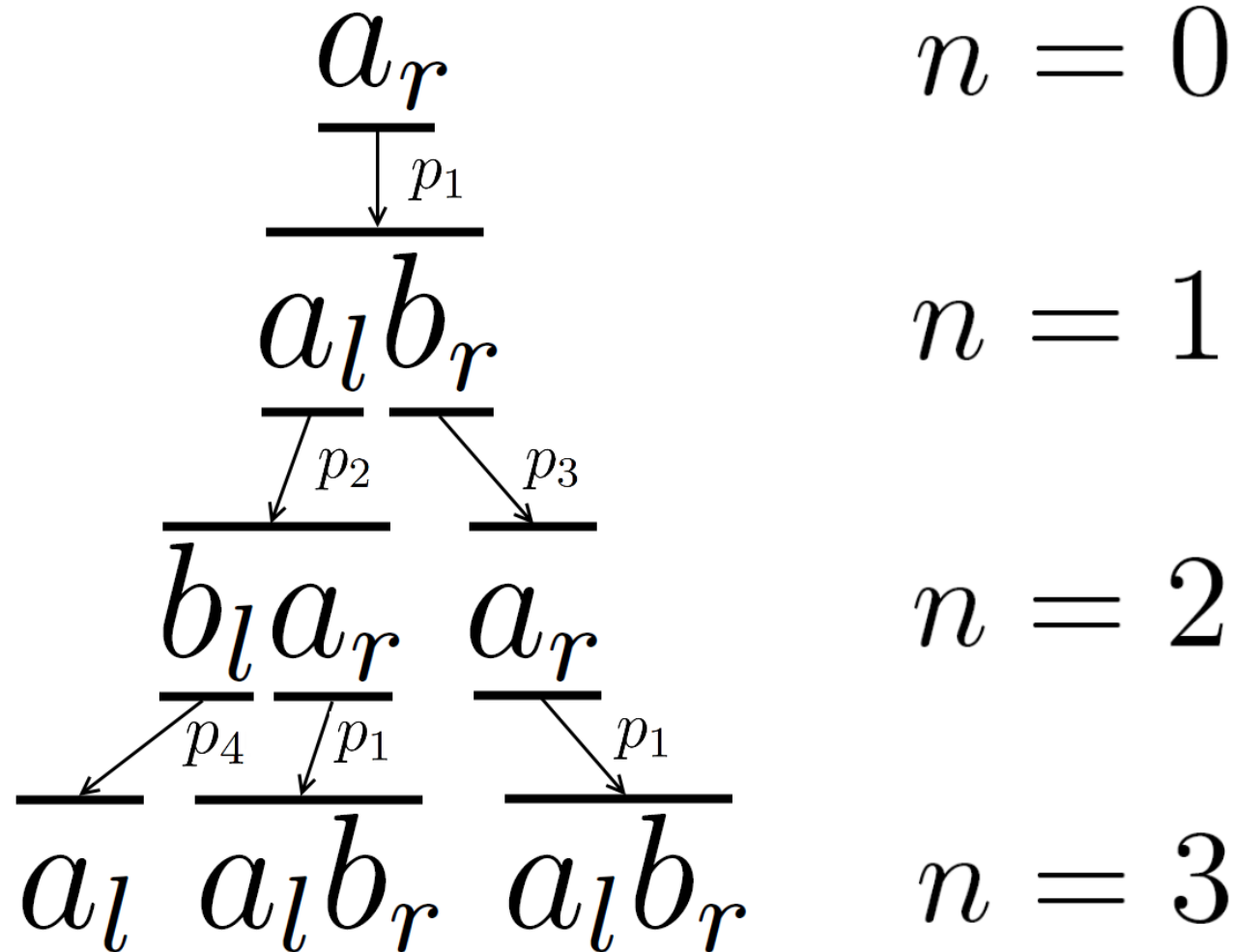
P besteht aus:

$$p_1 : a_r \rightarrow a_l b_r$$

$$p_2 : a_l \rightarrow b_l a_r$$

$$p_3 : b_r \rightarrow a_r$$

$$p_4 : b_l \rightarrow a_l$$



2. L-Systeme - parametrische L-Systeme

2.2 Parametrische L-Systeme

- Erweiterung der D0L-Systeme
- Verwendung von parametrischen Wörtern:

$A(a_1, \dots, a_n)$: Parametrisches Wort mit $A \in V$ und $a_1, \dots, a_n \in E(\Sigma)$

Σ : Menge formaler Parameter

$E(\Sigma)$: Arithmetischer Ausdruck

-

3 Space Colonization Algorithmus

3.1 TODO

3. Space Colonization Algorithmus - TODO

- TODO

4 Implementierung

- TODO

4. Implementierung - TODO

TODO

- TODO

5 Ergebnisse

5. Ergebnisse - TODO

5.1 TODO

6 Zusammenfassung und Ausblick

6. Zusammenfassung - TODO

6.1 TODO

Literatur

- [Bak] BAKER, MARTIN JOHN: *Maths - Angle between vectors*.
<http://www.euclideanspace.com/maths/algebra/vectors/angleBetween/index.htm>.
- [Bal98] BALZERT, HELMUT: *Lehrbuch der Software-Technik : Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Spektrum Akademischer Verlag GmbH, 1998.
- [Bec05] BECKER, PETE: *Working Draft, Standard for Programming Language C++*, 2005. <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2005/n1905.pdf>.
- [Blo85] BLOOMENTHAL, JULES: *Modeling the mighty maple*. Computer Graphics Laboratory, New York Institute of Technology, Old Westbu-

- ry, New York, 1985. <https://pdfs.semanticscholar.org/00d3/4582edd116a23d4d574ad2c90e9ebf01d74d.pdf>.
- [DL05] DEUSSEN, OLIVER und BERND LINTERMANN: *Digital Design of Nature - Computer Generated Plants and Organics*. Springer-Verlag Berlin Heidelberg 2005, 2005.
- [EKH10] EBERHARDT, HENNING, VESA KLUMPP und UWE D. HANEBECK: *Density Trees for Efficient Nonlinear State Estimation*, 2010. http://isas.uka.de/Publikationen/Fusion10_EberhardtKlumpp.pdf.
- [Eng] *Unreal Engine Documentation : Engine Features*. <https://docs.unrealengine.com/latest/INT/Engine/index.html>.
- [FGR] FINK, PROF. DR. SIEGFRIED, JÖRG GRÜNER und DR. CHRISTIAN RABE: *Skript zum Kernblock „Forstbotanik und Baumphysiologie II “ – Forstbotanischer Teil*. <https://www.forstbotanik.uni-freiburg.de/Lehre/Skripten/Skript%20Forstbotanik%20II>.

7. Literatur

- [Gre] *Green One - A landmark render of XfrogPlants by Jan Walter Schliep.* http://xfrog.com/gallery/landscapes/green01_big-1600small.jpg.php.
- [GSJ04] GOLDMAN, RON, SCOTT SCHAEFER und TAO JU: *Turtle Geometry in Computer Graphics and Computer Aided Design*, 2004. <http://www.cs.wustl.edu/~taoju/research/TurtlesforCADRevised.pdf>.
- [LN02] LEFEBVRE, SYLVAIN und FABRICE NEYRET: *Synthesizing Bark*, 2002. <http://www-evasion.imag.fr/Publications/2002/LN02/bark.pdf>.
- [Lux14] LUX, PROF. DR. ANDREAS: *Algorithmen und Datenstrukturen - Vorlesungsskript Kapitel 4*, 2014.
- [Man83] MANDELBROT, BENOIT B.: *The Fractal Geometry of Nature*. W. H. Freeman and Company, 1983.

7. Literatur

- [PL90] PRUSINKIEWICZ, PRZEMYSŁAW und ARISTID LINDENMAYER: *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, eBook Auflage, 1990. <http://algorithmicbotany.org/papers/abop/abop.pdf>.
- [Proa] *Procedural Mesh Component in C++ : Getting Started*. https://wiki.unrealengine.com/Procedural_Mesh_Component_in_C%2B%2B:Getting_Started.
- [Prob] *Profiling, How To Count CPU Cycles Of Specific Blocks Of Your Game Code*. https://wiki.unrealengine.com/Profiling,_How_To_Count_CPU_Cycles_Of_Specific_Blocks_Of_Your_Game_Code.
- [Ran] *Unreal Engine 4 Documentation : Random Streams - Initial Seed*. <https://docs.unrealengine.com/latest/INT/Engine/Blueprints/UserGuide/RandomStreams/#initialseed>.
- [RFL⁺05] RUNIONS, ADAM, MARTIN FUHRER, BRENDAN LANE, PAVOL FEDERL, ANNE-GAËLLE ROLLAND-LAGAN und PRZEMYSŁAW PRUSIN-

7. Literatur

- KIEWICZ: *Modeling and visualization of leaf venation patterns*, 2005. <http://algorithmicbotany.org/papers/venation.sig2005.pdf>.
- [RLP07] RUNIONS, ADAM, BRENDAN LANE und PRZEMYSŁAW PRUSINKIEWICZ: *Modeling Trees with a Space Colonization Algorithm*, 2007. <http://algorithmicbotany.org/papers/colonization.egwnp2007.pdf>.
- [Sch14] SCHMITZ, PROF. DR. HEINZ: *Theoretische Informatik - Vorlesungsskript*, 2014.
- [STN16] SHAKER, NOOR, JULIAN TOGELIUS und MARK J. NELSON: *Procedural Content Generation in Games*. Springer International Publishing Switzerland 2016, 2016.
- [Sura] SURIDGE, JAYELINDA: *Modelling by numbers: Part One A: An introduction to procedural geometry*. http://www.gamasutra.com/blogs/JayelindaSuridge/20130903/199457/Modelling_by_numbers_Part_One_A.php.

7. Literatur

- [Surb] SURIDGE, JAYELINDA: *Modelling by numbers: Part Two A: The cylinder.* http://www.gamasutra.com/blogs/JayelindaSuridge/20130905/199626/Modelling_by_numbers_Part_Two_A.php.
- [TKSY] TOGELIUS, JULIAN, EMIL KASTBJERG, DAVID SCHEDL und GEORGIOS N. YANNAKAKIS: *What is Procedural Content Generation? Mario on the borderline* . <http://julian.togelius.com/Togelius2011What.pdf>.
- [Unra] *Unreal Engine Documentation : Content Examples.* <https://docs.unrealengine.com/latest/INT/Resources/ContentExamples/>.
- [Unrb] *Unreal Engine Documentation : Unreal Engine 4 Terminology.* <https://docs.unrealengine.com/latest/INT/GettingStarted/Terminology/index.html>.
- [Wha] *Unreal Engine Features.* <https://www.unrealengine.com/unreal-engine-4>.